Report No. NADC-91004-60

AD-A234 990

# FLIGHT CONTROL LAW SYNTHESIS USING NEURAL NETWORK THEORY

Robert D. DiGirolamo
Shawn T. Donley
Air Vehicles and Crew Systems Technology Department (Code 6012)
NAVAL AIR DEVELOPMENT CENTER
Warminster, PA 18974-5000

OCTOBER 1990

INTERIM REPORT
Task No. R00N0000
Project No. 0
Work Unit No. HA650
Program Element No. 0601152N

*Approved for public release; distribution is unlimited*

Prepared for
NAVAL AIR DEVELOPMENT CENTER
Warminster, PA. 18974-5000

## NOTICES

REPORT NUMBERING SYSTEM — The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Officer or the Functional Department responsible for the report. For example: Report No. NADC-88020-60 indicates the twentieth Center report for the year 1988 and prepared by the Air Vehicle and Crew Systems Technology Deartment. The numerical codes are as follows:

| CODE | OFFICE OR DEPARTMENT |
|---|---|
| 00 | Commander, Naval Air Development Center |
| 01 | Technical Director, Naval Air Development Center |
| 05 | Computer Department |
| 10 | AntiSubmarine Warfare Systems Department |
| 20 | Tactical Air Systems Department |
| 30 | Warfare Systems Analysis Department |
| 40 | Communication Navigation Technology Department |
| 50 | Mission Avionics Technology Department |
| 60 | Air Vehicle & Crew Systems Technology Department |
| 70 | Systems & Software Technology Department |
| 80 | Engineering Support Group |
| 90 | Test & Evaluation Group |

PRODUCT ENDORSEMENT — The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

Reviewed By: Gary J. Gross   Date: 1/18/91
Branch Head

Reviewed By: [illegible]   Date: 1/22/91
Division Head

Approved By: W. F. Smith   Date: 1/25/91
Director/Deputy Director

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



**1. AGENCY USE ONLY (Leave blank)**

**2. REPORT DATE**
1990 October 31

**3. REPORT TYPE AND DATES COVERED**
Interim (Oct. 1989 - Sept. 1990)

**4. TITLE AND SUBTITLE**
Flight Control Law Synthesis
Using Neural Network Theory

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
DiGirolamo, Robert D.
Donley, Shawn T.

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Air Development Center (6012)
Warminster, PA 18974-5000

**8. PERFORMING ORGANIZATION REPORT NUMBER**
NADC-91004-60

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Naval Air Development Center
Warminster, PA 18974-5000

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for Public Release;
Distribution Unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT (Maximum 200 words)**

A commonly used technique for advanced fighter aircraft control law development involves a lengthy process of linearizing the aircraft model and calculating many control system gains via conventional linear methods. This process must be repeated for a number of trim points within the flight envelope to achieve the aircraft stability and flying qualities mandated by various military specifications.

Neural networks have been used extensively in many applications such as pattern recognition and optimization because of their ability to create nonlinear mappings of continuous valued inputs through supervised learning.

This report outlines a concept which incorporates emerging neural network technology with present-day control theory to produce a system by which optimal controller gains can be automatically generated. The research completed to date and the results contained in this report are intended to provide a proof of concept by applying the neural network synthesis technique to some simplified linear and nonlinear examples.

**14. SUBJECT TERMS**
Multilayer feedforward connectionist network, optimization, gain scheduling, adaptive control, learning control and back-error propagation.

**15. NUMBER OF PAGES**
19

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | Same as Rpt |

NSN 7540-01-280-5500

Standard Form 298 (890104 Draft)
Prescribed by ANSI Std. 239-18
298-01

## Contents



## Figures

## Abbreviations, Acronyms and Symbols

| | |
|---|---|
| BEP | Back-Error-Propagation |
| DOF | Degrees of Freedom |
| $d_j$ | Desired neural network ouput |
| $e(t)$ | Error between aircraft response and desired response |
| $E_{as}$ | Average System Error |
| $E_{rms}$ | Root Mean Square Error |
| FCC | Flight Control Computer |
| LQG | Linear Quadratic Gaussian |
| LQR | Linear Quadratic Regulator |
| MRAC | Model Referencing Adaptive Controller |
| $n$ | Training cycle index |
| $netin_j$ | Net input to node j |
| $O_i$ | Output of node i in output layer |
| $P$ | Total number of design points |
| SISO | Single Input Single Output |
| $T$ | Period of response |
| $t$ | Time |
| $W_{ij}(n)$ | Weight connecting node i to node j at the $n^{th}$ training cycle |
| $x_a(t)$ | Aircraft response |
| $x_d(t)$ | Desired response |
| $X_j$ | Output of node j in a hidden or input layer |
| $\alpha$ | Momentum term |
| $\delta_j$ | Error term for node j |
| $\eta$ | Training constant |
| $\Theta_j$ | Bias term for node j |
| $\Theta_o$ | Temperature coefficient |

## Introduction

Control laws for modern fighter aircraft are typically developed using a "point" design approach. Figure 1 is an illustrated example of the complexity associated with this process. The engineer begins with a six degree of freedom (DOF) nonlinear model which completely describes the aerodynamics of the aircraft throughout its maneuvering envelope. The aircraft model normally evolves through extensive wind tunnel testing and computational fluid dynamic analyses and includes various uncertainties and inaccuracies. A baseline structure for the aircraft control system must then be developed which defines the necessary feedback parameters and compensations. From here, the engineer chooses a single trim (or stationary) point within the aircraft flight envelope. The control law design is effectively centered around this point. The aerodynamic model is then linearized at the trim point so that conventional linear control theory may be applied to generate gains and parameters for the baseline control structure. Other techniques such as Linear Quadratic Gaussian (LQG) or Linear Quadratic Regulator (LQR) may be used to generate controller gains which minimize a quadratic cost function. Once the controller gains have been generated, the engineer chooses a different trim point and again linearizes the aircraft model and generates the optimal controller gains at the new point. This method is iterated until a complete set of controller gains has been generated for several points within the flight envelope. Once the process has been completed, the engineer has essentially created a discrete nonlinear functional mapping from the aircraft envelope into the desired control system gains. This lengthy design technique, however, limits the total number of trim points the engineer is able to investigate for a given aircraft.

Implementation of the various flight control laws is done by "scheduling" the controller gains as a function of the of the trim point parameters (e.g. Mach number, altitude, wing sweep, etc.). In flight, these parameters are measured so that the proper controller gains can be found through standard look-up tables residing in the Flight Control Computer's (FCC) memory. However, the trim point parameters are measured as continuous variables and their values frequently fall between two discrete design points each of which corresponds to a particular set of controller gains. At this point the FCC must perform time consuming multi-variable interpolations/extrapolations to estimate the values for the controller gains.

In the past, neural networks have been used extensively in applications such as pattern recognition, functional synthesis and optimization [1],[2]. In control applications, they have been used as adaptive controllers and state estimators [3],[4],[5],[6]. Neural networks offer several advantages which may enhance the performance of gain scheduled controllers and simplify their development process.

> *For example, back-propagation networks can implement nonlinear mappings of continuous valued inputs through supervised learning. Neural network dynamics also display a fast adaptation rate to a large number of parameters and a uniform rate of convergence which is independent of network size. Neural networks exhibit natural robustness due to their generalization properties [from ref. 3].*

The objective of this research is to incorporate emerging neural network technology into a feasible concept which will simplify control system development by automating the design of controller gains. The benefits anticipated by developing this concept are twofold: 1) reduce time and personnel required to develop control laws and 2) increase the number of design points used in the flight envelope to improve controller performance.

## Concept

Our approach is to apply current neural network concepts to synthesize a nonlinear functional relationship between aircraft trim point parameters and the desired control system gains

for a fixed-structure control law. The concept, as illustrated in figure 2, compares the aircraft responses with predetermined desired response models in the neural control stage. The neural control stage "grades" how well the aircraft/controller is responding with respect to the desired response models. The grade is then used to adjust the neural network so that new controller gains may be produced which enable the aircraft/controller to achieve responses more closely matched to the desired responses. This cycle is performed repeatedly at each arbitrary trim point until the "grade" falls within allowable tolerances.

Figure 3 is a detailed description of the flight control law synthesis concept in block diagram form. The following is a brief description of each major component:

1) The Aircraft Model is a complete set of mathematical equations (nonlinear differential) which describe the motion of the aircraft in one or more of its axes. Aerodynamic data which reflect coefficients of lift, drag and moments for the aircraft at different flight conditions and configurations accompany the equations of motion.

2) The Fixed-Structure Feedback Controller contains the control laws which, in combination with the correct gains, provide the aircraft with the proper handling qualities and stability margins. Control laws are the mathematical equations which relate the pilot input commands and sensed aircraft parameters to control surface deflections. The fixed structure implies that input commands and sensed parameters are always present within the control laws but, their "weights" or gains (i.e. the degree to which they affect the control law) may vary as a function of the aircraft's flight condition.

3) The Performance Models are a set of second order linear transfer functions which are intended to exemplify the desired maneuvering characteristics of the aircraft/controller. The performance models are designed in accordance with the frequency and damping requirements listed in MIL-F-8785C "Military Specification for Flying Qualities of Piloted Airplanes." The frequency and damping requirements are based on classification (e.g. fighter, transport, etc.), category and desired flying quality level for the specific aircraft.

4) The Test Input is the pilot commanded input to both the aircraft and the performance model. Various test inputs are used to excite different modes in the aircraft model (e.g. pitch doublet is used to excite short period mode).

5) The Cost Functional or performance index is the mechanism used to determine the "mismatch" between the output of the performance models and the actual response of the aircraft at a given trim point. Although there are many mathematical relationships which may be used for the cost functional [7], a common one in control system design is the root mean square error ($E_{rms}$) given by the equation:

$$E_{rms} = \left[ \frac{1}{T} \int^{T} e^2(t)\,dt \right]^{\frac{1}{2}} \qquad (1)$$

Where $e^2(t) = [x_d(t) - x_a(t)]^2$ is the square error between the aircraft response $x_a(t)$ and the desired response $x_d(t)$ provided by the performance models over the period, T. If the performance index is averaged over all selected design points P in the envelope, the average system error ($E_{as}$) is obtained:

$$E_{as} = \sum_{P}(E_{rms})/P \qquad (2)$$

The $E_{as}$ has been previously referred to as the "grade" of the aircraft controller.

6) The Neural Network chosen for the control law synthesis concept is the multilayer feedforward perceptron network. This type of network is illustrated schematically in figure 4 and is typically composed of sets of nodes (processors) arranged in layers. The output of each node in the previous layer is then connected to the input of each node in the subsequent layer through an adjustable weight known as the synaptic weight. Layers between the output layer and the input layer are called hidden layers. The adjustable weights are intended to amplify, attenuate or inhibit transmission of nodal outputs from one layer to the next. The net input to each node in a layer is the sum of the weighted outputs of every node in the previous layer (equation 3) with the exception of the input layer [1]. A nodal output is then determined by the *activation function* of the particular node which maps its net input plus a node "bias" into the output of the node. Although it is not a requirement, all nodes within a network typically possess the same activation function. For the case of our multilayer network, an activation function known as the sigmoid function, given by equation (4) was chosen.

$$netin_j = \Sigma W_{ij} O_i \qquad (3)$$

$$O_j = 1/[1 + e^{-(netin_j + \Theta_j)/\Theta_0}] \qquad (4)$$

Where $W_{ij}$ is the synaptic weight connecting the output of $i^{th}$ node in layer N-1 to the input of node j in layer N. $\Theta_j$ is the bias input and $\Theta_0$ is the temperature coefficient used to modify the shape of the sigmoid at node j.

The neural network's main function in the control law synthesis concept is to develop for the aircraft controller a set of optimal gains for each design point in the envelope. Trim point parameters are furnished as inputs and the network transmits gains to the aircraft controller via its output layer.

7) The Training Algorithm is the vehicle by which synaptic weights and node biases of a particular neural network can be determined so that the network provides a specific functional mapping. The training algorithm can be thought of as a neural network "designer." Back-Error-Propagation (BEP) was one of the first effective training algorithms for multilayer perceptron networks and is still used in many applications because of its straight-forward implementation. BEP in the form used in this study was introduced by Rumelhart, Hinton and Williams [8],[9] in 1986. BEP learning is an iterative gradient algorithm that modifies the synaptic weights throughout a neural network to minimize the error between the actual output of the network and its desired output which is determined by the user [10],[11]. For the concept presented in this study, the neural network outputs are the aircraft controller gains which are not known apriori to training. Therefore, the algorithm must be modified slightly to remain useful in this application.

The desired synaptic weights for the network are achieved by making incremental changes to the weights according to the following equation:

$$W_{ij}(n+1) = W_{ij}(n) + \eta \delta_j X_i \qquad (5)$$

Where $W_{ij}(n)$ is the old synaptic weight, $W_{ij}(n+1)$ is the new weight and $\eta$ is the training constant. $O_i$ is the output of node i in a hidden layer (or the $i^{th}$ input $I_i$, when node j lies in the first hidden layer). $\delta_j$ is the error or gradient term for node j. If node j is an output node, then

$$\delta_j = O_j(1 - O_j)(-\delta E_{as}/\delta O_j) \qquad (6)$$

In the standard implementation of the BEP algorithm, the term $-\delta E_{as}/\delta O_j$ in equation 6 would be replaced by the error between the desired output and the actual network output $(d_j - O_j)$. As mentioned before, the desired output is unknown in our application but, the gradient of the average system error (eq. 2) with respect to the network outputs $(\delta E_{as}/\delta O_j)$ can be determined numerically.

If node j lies within the hidden layers of the network, then

$$\delta_j = X_j(1 - X_j)\sum_k \delta_k W_{jk} \ , \qquad (7)$$

where k is over all nodes in the layer directly following node j.

Using the recursive equation (5) with equations (6) and (7), the algorithm starts at the output nodes O, and works its way back through the successive hidden layers. Node biases are adapted in a nearly identical fashion by assuming they are connection weights from nodes with constant valued outputs.

Finally, a *momentum term* is added to equation (5) to accelerate convergence of the algorithm to the desired solutions [1],[10],[11]. The term suggests that the synaptic weight change which occurs from training cycle n to n+1 is somewhat related to the previous weight change from training cycle n-1 to n. The proportionality constant $\alpha$ determines the amount of the previous weight change that will be used. Incorporating the momentum term, equation (5) becomes

$$\begin{aligned} W_{ij}(n+1) &= W_{ij}(n) + \eta\delta_j X_i \\ &\quad + \alpha[W_{ij}(n) - W_{ij}(n-1)] \ , \end{aligned} \qquad (8)$$

where $0 < \alpha < 1$. Equation (8) is the expanded equation implemented in our training algorithm.

The flight control law synthesis concept as shown in figure 3 is very similar in structure to the popular adaptive control technique known as the Model Referencing Adaptive Controller (MRAC). Both the MRAC and the neural network concept described in this report use a performance model (or command model) and an adaptation scheme to achieve controller gain adjustments. There are however, some major distinction between the two approaches:

1) The MRAC operates *on-line* (i.e. while the aircraft is in flight) and therefore, must provide stability as well as proper flying qualities from the onset. On the other hand, the neural network concept described herein operates *off-line* (as a control designer) and can learn by experience which gains stabilize the aircraft and provide adequate handling characteristics.

2) The neural network incorporates memory into the concept in the form of synaptic weights. This is an important difference because the MRAC must constantly adjust the controller gains as the aircraft progresses through the envelope. The neural network concept however, remembers the gains at a particular trim point and makes adjustments as needed. Once the network has arrived at the proper gains, the training algorithm essentially halts until an event occurs (e.g. change in aircraft dynamics) which produces an output error between the aircraft/controller and the performance model.

Operation of the flight control law synthesis concept is summarized in the following steps:

**Step 1.** **Initialize Network Weights and Biases.** Randomly set all synaptic weights and node biases to small values.

**Step 2.** **Trim aircraft.** Set aircraft at trim point to begin design.

**Step 3.** **Calculate the Controller Gains.** Forward propagate the set of trim parameters (input) via equations 3 and 4 through the neural network and transfer the outputs to the aircraft controller.

**Step 4.** **Apply Test Input.** Initiate the test maneuver (e.g. pitch doublet) for the aircraft/controller and the performance model

**Step 5.** **Evaluate Performance of Controller.** Calculate the root mean square error ($E_{rms}$ - eq. 1) between the aircraft and performance model.

**Step 6.** **Repeat Steps 2 Through 5.** Repeat these steps for each trim point in the design envelope.

**Step 7.** **Grade the Neural Network.** With $E_{rms}$ calculations from step 5 determine the average system error ($E_{as}$ - eq. 2) and gradients for training.

**Step 8.** **Train the Neural Network.** Adjust the synaptic weights and node biases of the network (eq. 5 - 8) using BEP to reduce the $E_{as}$.

**Step 9.** **Repeat Procedure.** Repeat steps 2 through 7 until the $E_{as}$ has fallen below a prescribed tolerance.

## RESULTS

The flight control law synthesis concept and neural network algorithms aforementioned were coded in software and numerous simulations were performed to examine the effectiveness of this approach. As an initial effort, very simplified models and control structures replaced the aircraft model and flight control system. In addition, the performance models were also replaced by arbitrary models not intended in any way to describe actual aircraft handling characteristics as in Mil-F-8785C. This section contains some of the results of these computer simulations which were designed to support the basic ideas of the neural network/flight control law synthesis concept.

The first simulation used a second order Single Input Single Output (SISO) linear system in place of the aircraft model. The transfer function of the model is given by equation (9).

$$H_1(s) = 1/(s^2 + 2s + 1) \qquad (9)$$

This transfer function represents a critically damped system with a damping ratio of 1.0 and a natural frequency of 1.0 rad/sec. The control structure for this example is illustrated in figure 5. The controller provides proportional output and rate feedback via gains $K_1$ and $K_2$ and a proportional input gain $K_3$. An arbitrary SISO linear system has been chosen as the performance model for this example with a transfer function given by:

$$H_2(s) = 4/(s^2 + 2.82s + 4) \qquad (10)$$

The simulation was configured according to the following specifications:

| | | |
|---|---|---|
| Test Input | : | unit step |
| Plant Model | : | $H_1(s)$ (eq. 9) |
| Controller | : | figure 5;<br>3 design points* |
| Performance Model | : | $H_2(s)$ (eq. 10) |
| Cost Functional | : | $E_{rms}$ (eq. 1)<br>$E_{as}$ (eq. 2) |
| Training Algorithm | : | back-error propagation |
| Neural Network | : | 2 inputs (plant initial conditions)<br>3 hidden layers<br>layer 1 - 10 nodes<br>layer 2 - 12 nodes<br>layer 3 - 15 nodes<br>3 outputs (gains) |

*Note that a set of three different initial condition vectors $[x_1(0)\ x_2(0)]^T$ were used as inputs to the neural network and therefore, formed three design points for the controller.

Two separate trials were run for the first simulation and the initial controller gains were changed between trials. For each of the trials, the initial conditions and gains were set to:

Trial #1:
($x_1$ = 0.0, $x_2$ = 0.0) -> ($K_1$ = 0.0, $K_2$ = 0.0, $K_3$ = 1.0)
($x_1$ = 0.1, $x_2$ = 0.3) -> ($K_1$ = 0.0, $K_2$ = 0.0, $K_3$ = 1.0)
($x_1$ = 0.5, $x_2$ = 0.75) -> ($K_1$ = 0.0, $K_2$ = 0.0, $K_3$ = 1.0)

initial cost: 0.1881

Trial #2:
($x_1$ = 0.0, $x_2$ = 0.0) -> ($K_1$ = 2.0, $K_2$ = 5.0, $K_3$ = 5.0)
($x_1$ = 0.1, $x_2$ = 0.3) -> ($K_1$ = 2.0, $K_2$ = 5.0, $K_3$ = 5.0)
($x_1$ = 0.5, $x_2$ = 0.75) -> ($K_1$ = 2.0, $K_2$ = 5.0, $K_3$ = 5.0)

initial cost: 0.152

The gains $K_1$, $K_2$ and $K_3$ were adjusted dynamically by the neural network throughout the simulation so that the plant/controller response began to match more closely the response of the performance model in each successive training cycle. The results of both simulation trials are listed below:

Trial #1:
($x_1$ = 0.0, $x_2$ = 0.0) -> ($K_1$ = - 0.226, $K_2$ = 1.495, $K_3$ = 2.443)
($x_1$ = 0.1, $x_2$ = 0.3) -> ($K_1$ = - 0.226, $K_2$ = 1.492, $K_3$ = 2.438)
($x_1$ = 0.5, $x_2$ = 0.75) -> ($K_1$ = - 0.225, $K_2$ = 1.488, $K_3$ = 2.431)

training cycles: 200
final cost: 0.03876
performance improvement: 79.39%

Trial #2: ($x_1$ = 0.0, $x_2$ = 0.0) -> ($K_1$ = 1.386, $K_2$ = 3.611, $K_3$ = 4.645)
($x_1$ = 0.1, $x_2$ = 0.3) -> ($K_1$ = 1.385, $K_2$ = 3.610, $K_3$ = 4.646)
($x_1$ = 0.5, $x_2$ = 0.75) -> ($K_1$ = 1.385, $K_2$ = 3.610, $K_3$ = 4.648)

training cycles: 55
final cost: 0.009869
performance improvement: 93.51%

Figures 6 and 7 show the initial and final responses for each trial run at the first initial condition. The desired responses are identified by a dashed line and the actual plant/controller responses by the solid line. It should be noted that trial #2 arrived at a better solution quicker th. : trial #1. This result is primarily due to the choice of initial gains for each trial run. The first set of initial gains caused the network and training routine to progress to a local minimum at $K_1$ = -0.226, $K_2$ = 1.492, $K_3$ = 2.438). Whereas, the second set of initial gains caused the network to progress to a lower minimum at $K_1$ = 1.385, $K_2$ = 3.610, $K_3$ = 4.646). It should also be noticed that the network arrived at essentially the same gains for each initial condition. This is a correct result because it can be shown that a linear plant (as in the first simulation) requires only a linear control system (i.e. one set of gains for all conditions) to produce the required response.

The final set of simulations also used a second order SISO system in place of the aircraft model. However, a n~nlinearity was incorporated into the model to provide the neural network/flight control lav 'nthesis concept with a more realistic problem. Nonlinearities in plant dynamics can have many forms. A rate limiter was included in this particular example to represent a typical nonlinear system. The linearized transfer function of the model is given by equation (11).

$$H_1(s) = 4/(s^2 + 1.2s + 4) \tag{11}$$

This transfer function represents a underdamped system with damping ratio of 0.3 and a natural frequency of 2.0 rads/sec. The control structure and performance model for this example are identical to those presented in the previous example.

This simulation was configured according to the following specifications:

| | | | |
|---|---|---|---|
| Test Input | : | unit step | |
| Plant Model | : | nonlinear;<br>rate limited | (eq. 11) |
| Controller | : | figure 5;<br>3 design points | |
| Performance Model | : | $H_2(s)$ | (eq. 10) |
| Cost Functional | : | $E_{rms}$<br>$E_{as}$ | (eq. 1)<br>(eq. 2) |
| Training Algorithm | : | back-error propagation | |
| Neural Network | : | 2 inputs (plant initial conditions)<br>2 hidden layers<br>layer 1 - 5 nodes<br>layer 2 - 10 nodes<br>3 outputs (gains) | |

As in the first simulation example, two separate trial run were made and the initial controller gains were changed between trials. For the each of the trials, the initial conditions and gains were set to:

| | | | |
|---|---|---|---|
| Trial #1: | $(x_1 = 0.0, x_2 = 0.0)$ | -> | $(K_1 = 0.0, K_2 = 0.05, K_3 = 1.0)$ |
| | $(x_1 = 0.1, x_2 = 0.3)$ | -> | $(K_1 = 0.0, K_2 = 0.05, K_3 = 1.0)$ |
| | $(x_1 = 0.5, x_2 = 0.75)$ | -> | $(K_1 = 0.0, K_2 = 0.05, K_3 = 1.0)$ |

initial cost: 0.09864

| | | | |
|---|---|---|---|
| Trial #2: | $(x_1 = 0.0, x_2 = 0.0)$ | -> | $(K_1 = 0.9, K_2 = 2.05, K_3 = 2.75)$ |
| | $(x_1 = 0.1, x_2 = 0.3)$ | -> | $(K_1 = 0.9, K_2 = 2.05, K_3 = 2.75)$ |
| | $(x_1 = 0.5, x_2 = 0.75)$ | -> | $(K_1 = 0.9, K_2 = 2.05, K_3 = 2.75)$ |

initial cost: 0.1080

Again, the gains $K_1$, $K_2$ and $K_3$ were adjusted dynamically by the neural network throughout the simulation so that the plant/controller response began to match more closely the response of the performance model in each successive training cycle. The results of both simulation trials are listed below:

| | | | |
|---|---|---|---|
| Trial #1: | $(x_1 = 0.0, x_2 = 0.0)$ | -> | $(K_1 = 0.754, K_2 = 1.279, K_3 = 2.313)$ |
| | $(x_1 = 0.1, x_2 = 0.3)$ | -> | $(K_1 = 0.741, K_2 = 1.271, K_3 = 2.293)$ |
| | $(x_1 = 0.5, x_2 = 0.75)$ | -> | $(K_1 = 0.724, K_2 = 1.262, K_3 = 2.270)$ |

training cycles: 100
*final cost*: 0.04415
performance improvement: 55.24%

| | | | |
|---|---|---|---|
| Trial #2: | $(x_1 = 0.0, x_2 = 0.0)$ | -> | $(K_1 = 1.033, K_2 = 2.593, K_3 = 3.593)$ |
| | $(x_1 = 0.1, x_2 = 0.3)$ | -> | $(K_1 = 1.039, K_2 = 2.616, K_3 = 3.627)$ |
| | $(x_1 = 0.5, x_2 = 0.75)$ | -> | $(K_1 = 1.052, K_2 = 2.654, K_3 = 3.686)$ |

training cycles: 50
final cost: 0.0443
performance improvement: 58.98%

Figures 8 and 9 show the initial and final responses for each trial run of the nonlinear simulation example. The desired responses are again identified by a dashed line and the actual plant/controller responses by the solid line. In this example, the concept concluded with roughly the same amount of success at both sets of initial gains. There is an initial mismatch between the desired and actual responses in both cases due to the rate limiting imposed by the plant dynamics. It is interesting to note that the neural network did not attempt to drive up the feedback gains of the controller to overcompensate for this nonlinear rate limit. The network, however, realized the physical limitations of the problem and attempted to provide the best possible solution.

## CONCLUSIONS

Based on an analysis of the results obtained from the computer simulations and discussed in the previous section, some concluding statements can be made about our neural network/flight control law synthesis concept:

1) The neural network, working within our concept, was able to produce feasible controller gains which improved the plant/controller response with respect to the performance model.

2) The neural network was able to accomplish (1) without any prior knowledge of the plant, controller or performance model dynamics.

3) The neural network did not supply controller gains which would cause the plant/controller to become unstable or diverge from the performance model.

4) The neural network provided better results faster when the initial gains were close to optimum.

5) The neural network will not always converge to an optimal solution. The network gives no indication that it has arrived at either an optimal or sub-optimal solution.

6) The neural network did not attempt to overcompensate for limitations imposed by nonlinearities within the plant (e.g. continually increase a feedback gains when the plant/controller is rate saturated).

7) The neural network did not require specific knowledge regarding the source or location of the error (i.e. steady state error, peak overshoot error, etc.) but was able to increase the plant/controller performance based on $E_{rms}$ information alone.

In light of the preceding conclusions, a neural network operating in an environment such as the one describe herein may offer several advantages over conventional controller design methods. Such advantages include: 1) a simplified design procedure due to the networks ability to produce optimal gains without specific knowledge of aircraft dynamics, 2) a more accurate controller design due to the networks ability to handle a large number of parameters and many design points and 3) a more robust controller due to the generalization properties of the network. However, it should be noted that the neural network field is far from mature and many sources of uncertainty still exist which inhibit a complete understanding of their operation within application such as the control law synthesis concept. And although reasonable results were obtained for the selected linear and nonlinear examples herein, a great effort and challenge still remains to apply such a concept to the highly complex dynamics of an aircraft model.

## FUTURE WORK

In the future, work of this nature shall progress in the following direction:

1) Software integration of a nonlinear three degree-of-freedom aircraft simulation model. The model shall include force and moment equations for the longitudinal axis of an F/A-18 aircraft with aerodynamic data for the full maneuvering envelope.

2) Develop a set of performance models via Mil-F-8785C for the longitudinal axis of a Class IV, Category A high maneuverability fighter/attack aircraft with Level I handling qualities.

3) Conduct experiments with 3-DOF aircraft model and the neural network/flight control law synthesis concept via repeated computer simulation and analysis.

4) Evaluate performance of aircraft and neural network generated controller.

5) Investigate potentials for improving concept and possibilities of implementing variations of this concept in an on-line adaptive/learning control scheme.

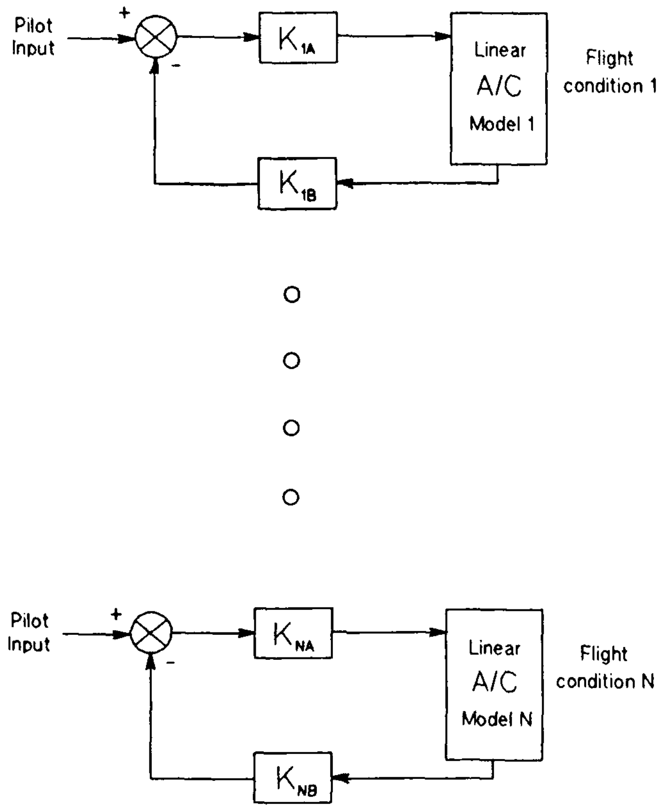


FIGURE 1: CONVENTIONAL POINT DESIGN

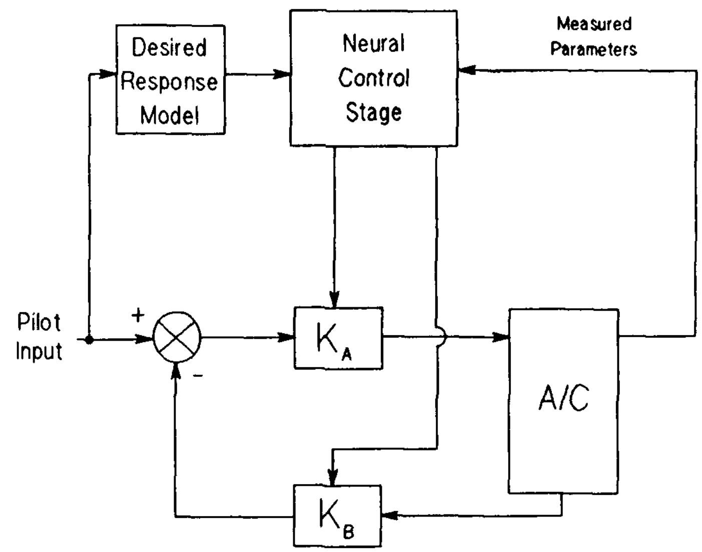


FIGURE 2: NEURAL NETWORK APPROACH

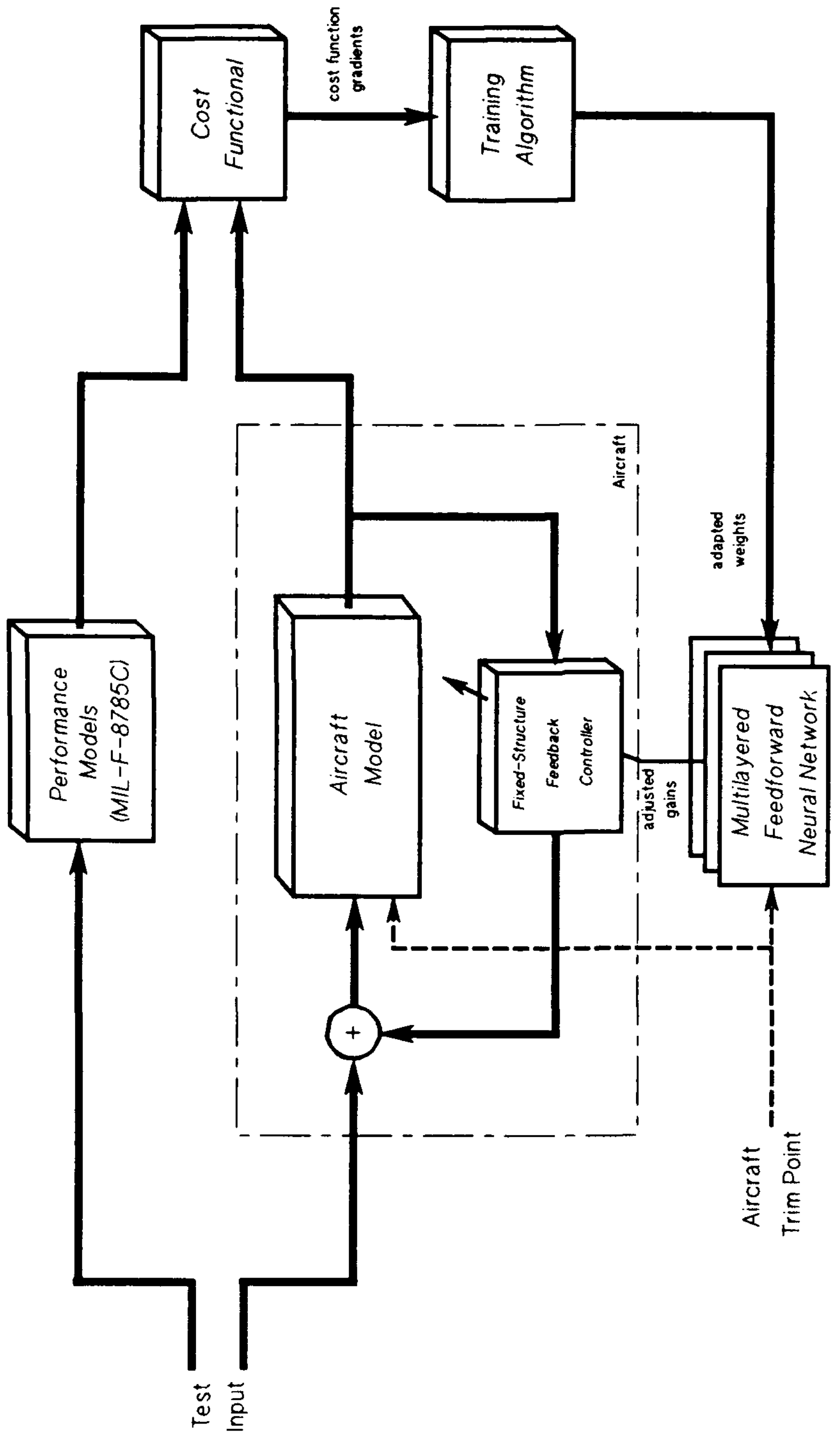


FIGURE 3: FLIGHT CONTROL LAW SYNTHESIS CONCEPT

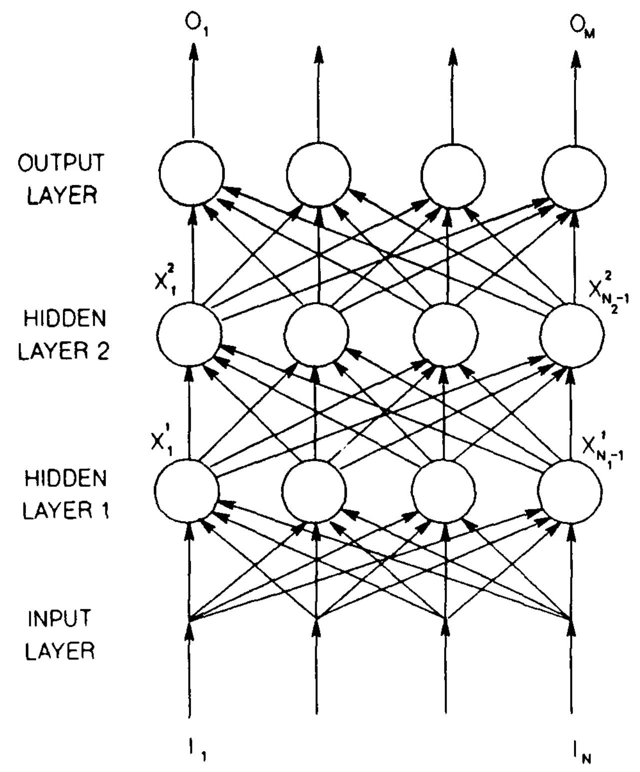


FIGURE 4: MULTILAYER FEEDFORWARD PERCEPTRON NETWORK

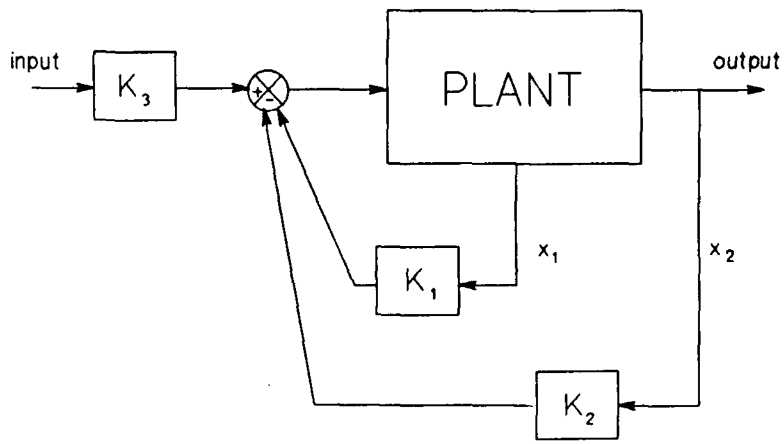


FIGURE 5: CONTROL STRUCTURE FOR LINEAR AND NONLINEAR EXAMPLES

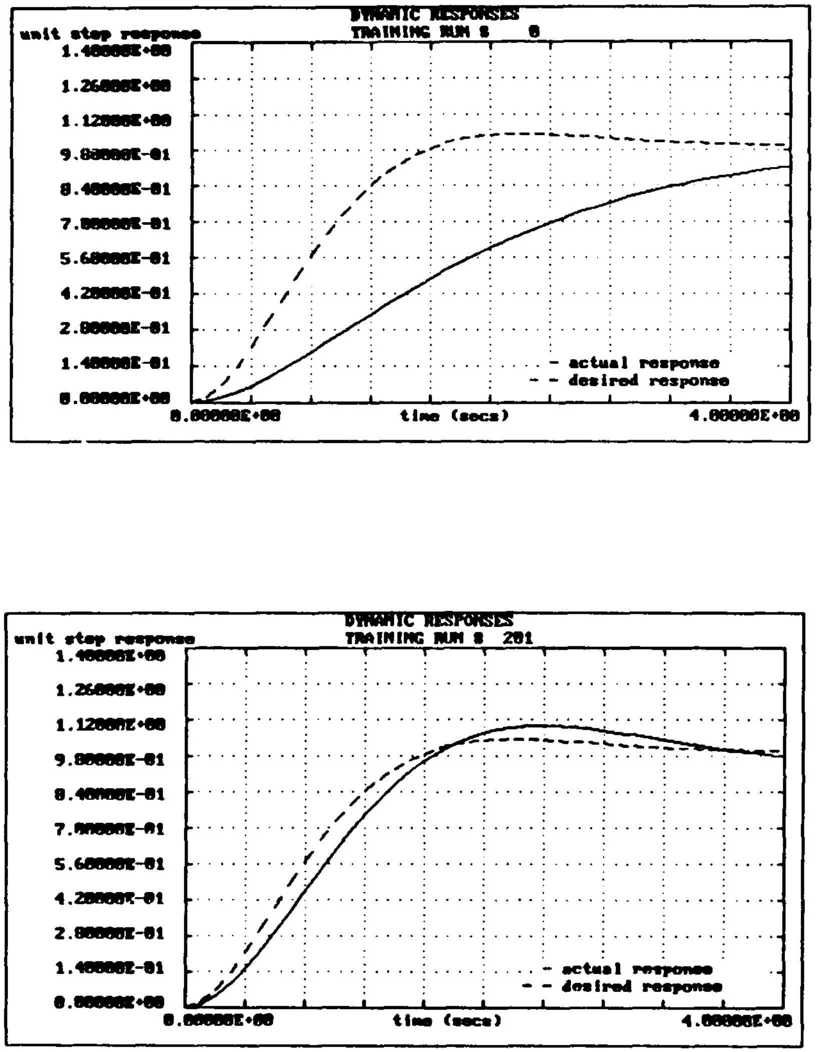


FIGURE 6: INITIAL AND FINAL RESPONSES FOR LINEAR EXAMPLE - TRIAL #1

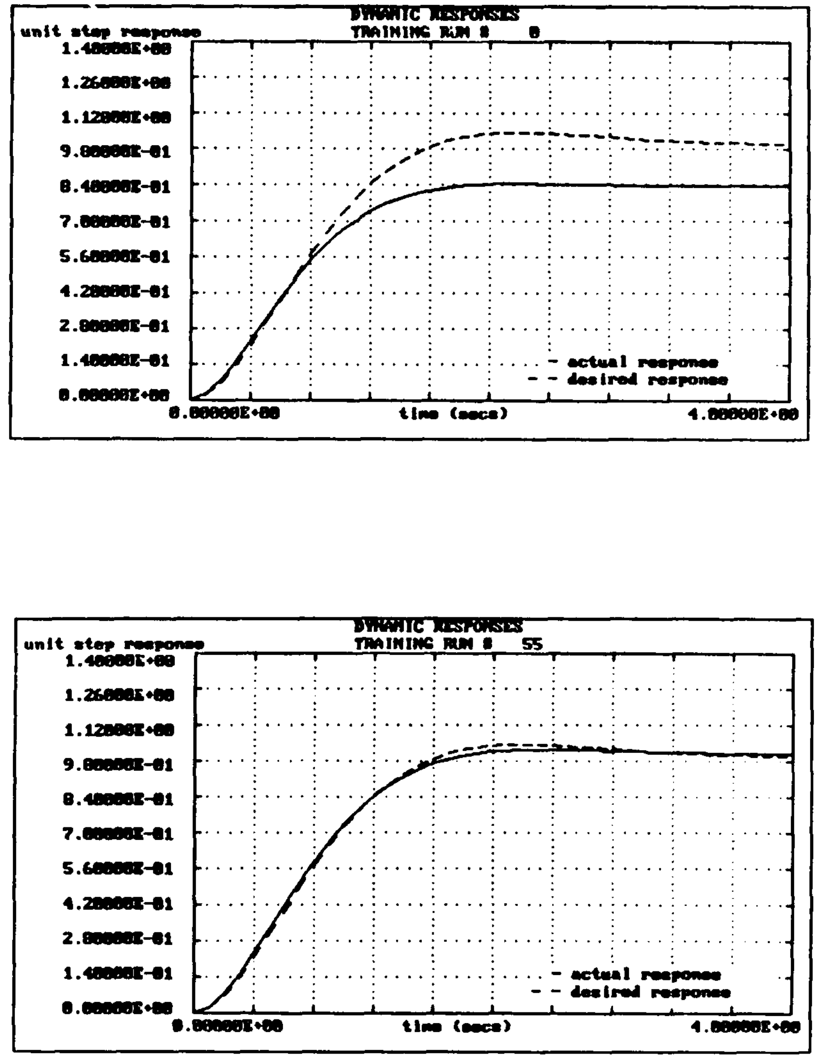


FIGURE 7: INITIAL AND FINAL RESPONSES FOR LINEAR EXAMPLE - TRIAL #2

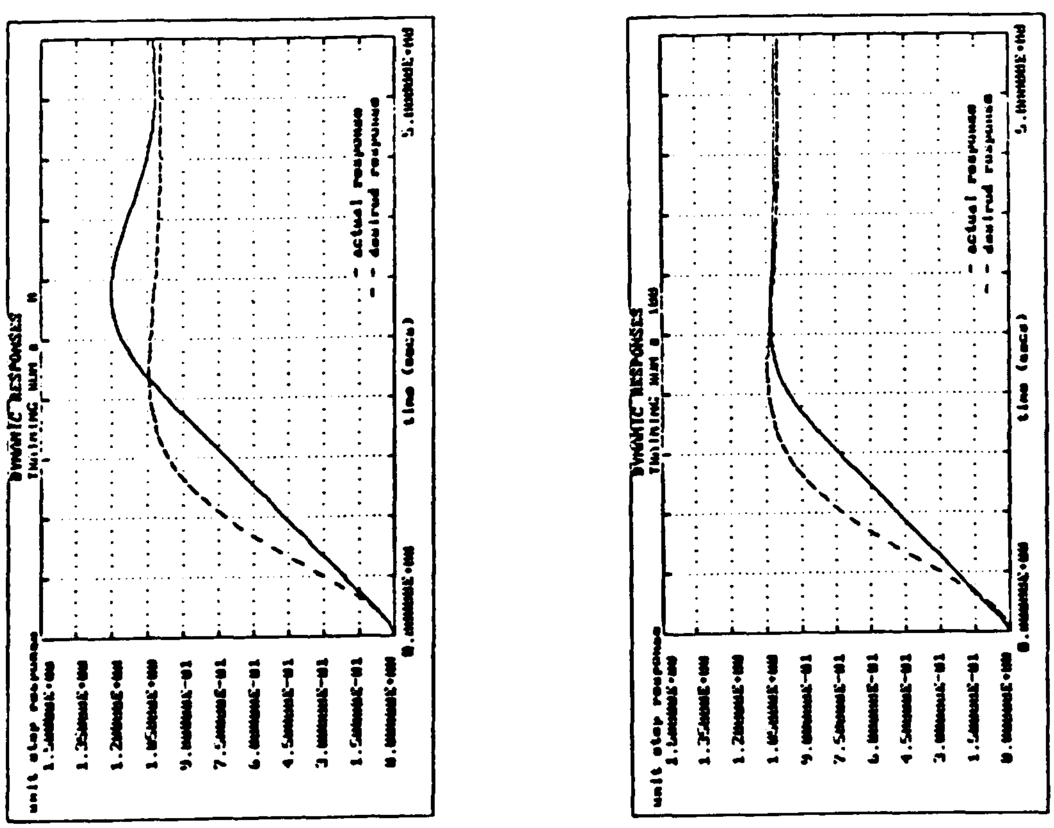


FIGURE 8: INITIAL AND FINAL RESPONSES FOR NONLINEAR EXAMPLE - TRIAL #1

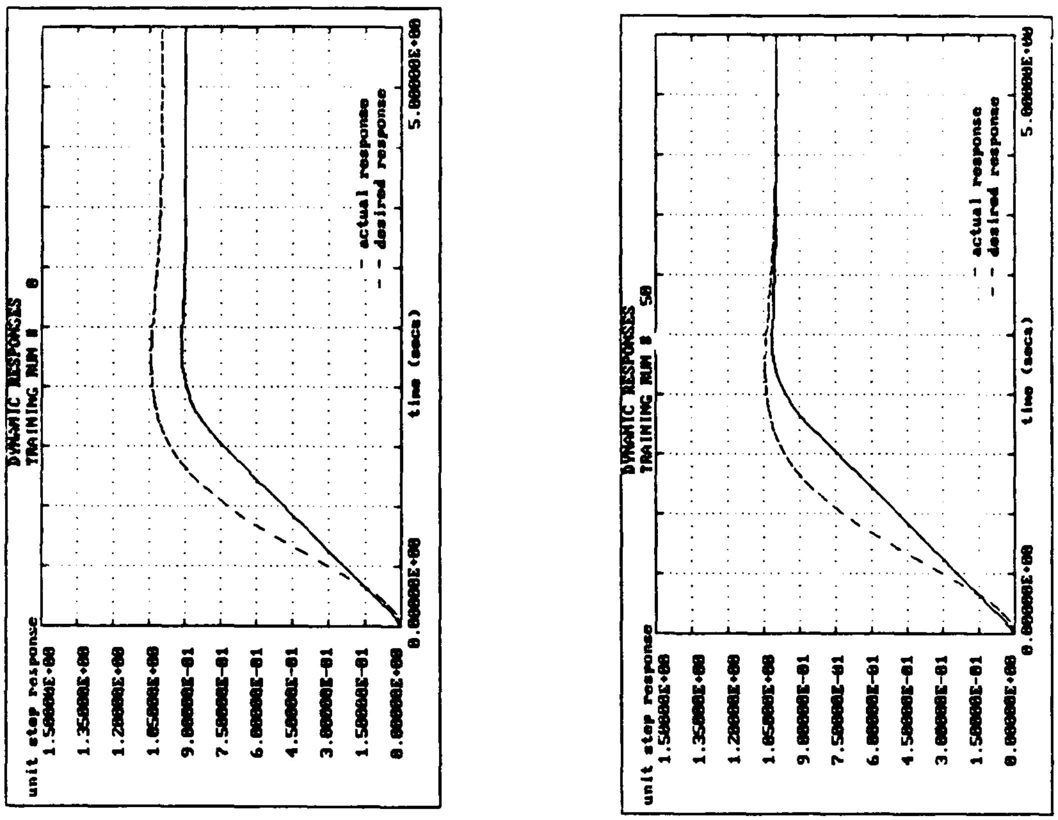


FIGURE 9: INITIAL AND FINAL RESPONSES FOR NONLINEAR EXAMPLE - TRIAL #2

REFERENCES

[1] Pao, Y.H., Adaptive Pattern *Recognition and Neural Networks,* Addison-Wesley Publishing Co., 1989.

[2] Hopfield, J.J. and D.W. Tank, *"Neural" Computation of Decisions in Optimization Problems,* Biological Cybernetics, 1985.

[3] Guez, A. and J. Selinsky, *A Trainable Neuromorphic Controller, Journal of Robotic Systems,* March-April 1988.

[4] Glanz, F.H., W.T. Miller, L.G. Kraft and E. An, *Survey of Neural Network Applications to Control,* University of New Hampshire Robotics Laboratory, 1988.

[5] Psaltis, D., A. Sideris and A. Yamamura, *Neural Controllers,* California Institute of Technology.

[6] Psaltis, D., A. Sideris and A. Yamamura, *A Multilayered Neural Network Controller,* IEEE Control Systems Magazine, April 1988.

[7] Hasdorff, L., *Gradient Optimization and Nonlinear Control,* John Wiley and Sons, 1976, pp 197-213.

[8] Rumelhart, D., G. Hinton, and R. Williams, *Learning Internal Representations by Error Propagation in Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* MIT Press, 1986.

[9] McClelland, J. and D. Rumelhart, *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs and Exercises,* MIT Press 1988.

[10] Koch, C. and I. Segev, eds., *Methods in Neuronal Modeling,* MIT Press, 1989, pp. 392-394.

[11] Lippman, R.P., *An Introduction to Computing with Neural Nets,* IEEE ASSP Magazine, April 1987.

[12] Grossberg, S., *Studies of the Mind and Brain,* Reidel Publishing Co., 1982.

## Distribution List

No. of Copies

Adminstrator
Defense Technical Information Center
Cameron Station, Building 5
Alexandria, VA 22314
Attn: DTIC-DDA-1 .......... 2

Commander
Naval Air Systems Command (AIR00D4)
Department of the Navy
Washington, DC 20361
(2 for retention)
(1 for Air 53014) .......... 3

NAVAIRDEVCEN
Warminster, PA 18974
(3 for Code 8131)
(11 for Code 6012) .......... 14